Attorney Docket: 2117P

UNITED STATES PATENT APPLICATION

FOR

MEMORY MANAGEMENT FOR PACKET SWITCHING DEVICE

Inventor:

Ofer Iny

Joseph A. Sawyer, Jr.
Sawyer Law Group LLP
2465 E. Bayshore Road, Suite 406
Palo Alto, California 94303

MEMORY MANAGEMENT FOR PACKET SWITCHING DEVICE

FIELD OF THE INVENTION

The present invention relates generally to packet switching devices and more particularly to memory management for packet switching devices.

BACKGROUND OF THE INVENTION

Multi-port packet switching devices by their nature require memory to temporarily buffer packets before transmission to their destination. Typically in a packet switching device a plurality of queues is required. At the minimum one queue per output port is required. Packet switching devices employ either an input, input output, output, or shared memory architectures or a combination of these. In most cases a bulk memory needs to be managed as a plurality of independent queues. A switching device may have a plurality of such memories. It is desirable to be able to manage such memory in the most efficient way as to facilitate a large number of queues and fast access with the minimum of resources. The present invention addresses such a need.

SUMMARY OF THE INVENTION

A method and system for managing memory in a packet switching device is disclosed. The method and system comprises managing the memory as a single FIFO when inserting packets and managing the memory as a plurality of FIFO queues when removing packets.

A memory management scheme in accordance with the present invention takes advantage of the nature of packet switching to give an efficient implementation of multiple

independent queues in a memory. The key observation that is made is that in a packet switch it is expected that packets may be stored in the memory for a time no greater than the time it takes to fill up the memory. If a packet is delayed due to extreme congestion then deleting that packet is an acceptable result.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a block diagram of a high performance switch block.

Figure 2 illustrates a block diagram of a shared memory switch block.

Figure 3 is a depiction of a memory after initialization.

Figure 4 is a depiction of the memory after 4 packets have been written in.

Figure 5 is a depiction of the memory after 5 packets have been written and the first packet deleted.

Figure 6 is a depiction of the memory after the second packet was removed.

Figure 7 is a diagram of the memory manager component.

DETAILED DESCRIPTION

The present invention relates generally to packet switching devices and more particularly to memory management for the packet switching device. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope

consistent with the principles and features described herein.

A system in accordance with the present invention utilizes a global input FIFO which manages the memory as a single FIFO when packets are inserted thereto and manages the memory as a plurality of FIFO queues when removing packets. To describe the global input FIFO, a memory manager maintains a Global FIFO Write Pointer (GFWP), and a Global FIFO Delete Pointer (GFDP). To describe the plurality of queues, the memory manager maintains Queue Tail Pointers, Queue Head Pointers, and Queue Link Pointers. Optionally the memory manager may maintain a Queue Size and/or a queue empty flag.

A packet's data is written into the memory sequentially starting at the GFWP. As the packet's data is written the GFWP advances. The packets in the memory are stored in a chronological sequential order starting from above the GFDP (oldest) to just below the GFWP (newest). In this way, the start address and the length of the packets in memory words describe a packet.

The memory manager constructs a database describing the global FIFOs memory's content as a plurality of queues using the queue tail pointers, queue head pointers, and queue link pointers. The GFDP precedes the GFWP by a preset value.

The GFDP advances when a packet's data is written into the memory to maintain the separation. If the GFDP points to an active start of packet (SOP) then a packet is scheduled to be removed from the queue to which that packet belongs. In this way if a packet remains in the switching device's memory longer then it takes to fill up the global FIFO then it is deleted from the memory. This resolves the single FIFO and multiple FIFO queues representation of the memory. The memory between the GFDP and the GFWP, called the empty zone, may not contain packets that belong to a queue.

The main advantages of this method are: no free memory management, packets are written in a sequential order and memory is efficiently used, packets are deleted if they stay in the memory longer then it takes to fill it up, it is conducive to queuing systems with a large number of queues in a limited amount of memory.

To describe the features of the present invention in more detail, refer now to the following discussion in conjunction with the accompanying figures.

Figure 1 illustrates a block diagram of a high performance switch block 10. Typical high performance switches require input and output packet queues. The switch block 10 includes a plurality of ingress controllers 12a-12n and a plurality of egress controllers 14a-14n. Each of the ingress controllers and egress controllers are coupled to an interconnect network 16. Each of the ingress controllers 12a-12n includes an input data memory 13, a memory manager 17 and an input controller 19. Each of the egress controllers 14a-14n includes an output data memory 23, a memory manager 25 and an output controller 27. Variable sized packets are stored in the input memory of the ingress controller 12a-12n. The input memory is organized as a plurality of FIFO queues. Typically there is at least one queue per output port and priority at the input. Packets are switched to the output port 14a-14n through the interconnect network 16. Packets are stored at the output port prior to transmission. Typically there are a plurality of FIFO queues at the output per input port and per priority.

Figure 2 illustrates a block diagram of a shared memory switch block 50. The switch block 50 comprises a plurality of switch port controllers 52a-52n which communicate with memory manager 54. The memory manager 54 in turn is in communication with a shared data memory 56. In this system, the memory is managed as a plurality of FIFO queues. Typically there is a plurality of queues per input-output pair and per priority. The memory manager 54

maintains a description of the memory as a plurality of FIFO queues accessible by the switch port controllers 52a-52h.

Figure 3 is a depiction of a memory which is managed by a memory manager after initialization. Referring to Figure 1, a sample memory 100 is shown with 16 words (Of course normal memories will have many more words). After initialization the global FIFO write pointer 112 is initialized to point to word 0 and the global FIFO delete pointer 114 is initialized to point to word 4. At this time there are no packets in the memory.

Figure 4 is a depiction of the memory 100 after 4 packets have been written in. The first packet is 3 memory words long. The start of packet (SOP) is written into word 1 of the memory and the end of packet (EOP) is written into word 3. Packets 2,3,4 are of size 2,4,1 respectively. After writing all the packets the GFWP is pointing to word 10 and the GFDP is pointing to word 14.

There are two active (non-empty) queues. Packets 1,4 are in queue 0 and packets 2,3 are in queue 1. Queue 0 is defined by; the Queue Head Pointer 0, which is pointing to word 0 where the SOP of packet 1 is, Queue Tail Pointer 0, which is pointing to word 9 where the SOP of the last packet in the queue is, and a link pointer, which points from packet 1 SOP to packet 4 SOP. Similarly, queue 1 is defined by: Queue Head Pointer 1, Queue Tail Pointer 1, and a Link Pointer.

Figure 5 is a depiction of the memory after 5 packets have been written and the first packet deleted. Referring to Figure 5, packet number 5 is added to the memory. As a result the GFWP is incremented to word 13 and the GFDP to word 1. As the GFDP is incremented it goes through word 0. When it points to word 0 we see that there is an active packet SOP that belongs to queue 0 (the mechanism to make this determination is explored further

below). The memory manager then forces a packet to be removed from queue 0. As a result, in this case, that same packet was removed from the memory. In a switching device this results in the discard of a packet. As a result, queue 0 head pointer **117** was reset to the link pointer's value.

Figure 6 is a depiction of the memory 100 after the second packet was removed. Referring to Figure 6, packet number 2 was removed from queue 1. In a switching device this may have happened due to the transmission of the packet or an "intelligent" discard. As a result, queue 1 head pointer 119 was reset to the link pointer's value.

Figure 7 is a diagram of the memory manager component 200. Referring to Figure 5, one implementation of the memory management, with values shown corresponding to memory configuration in Figure 2, has separately addressable; data memory, section queue ID memory, section link memory, queue head pointers memory, queue tail section pointers memory, and queue size memory. A section is defined as a block of memory that can have at most one start of packet word (SOPW). In the example shown it is assume that each data memory word may contain an SOPW.

The data memory contains packet data words. The section queue ID memory contains a flag indicating if an SOPW exist in the corresponding data section and if so the queue ID to which the SOPW belongs. The section link memory contains the link pointers to the next SOPW of the next packet in the queue and that packet's size. The queue head memory contains the pointers to the queue's last packet's SOPW and that packet's size. The queue tail section pointers memory contains a pointer to the section of the tail (youngest) packet's SOPW of the queue.

A way of determining when the GFDP points to an active packet is by maintaining a

corresponding word per memory section that contains the queue ID to which a possible SOPW belongs and possibly also a flag indicating if the section has an SOPW. When a packet start word is written, the corresponding section queue ID memory is written with the queue ID to which the packet belongs and possibly the SOP flag is set. If an SOP flag is maintained then, when a section is written without an SOPW the SOP flag is cleared. When the GFDP advances to a new section, this section queue ID is read. Then if the queue is not empty the head pointer of that queue is compared to the GFDP. If the two are equal then the packet is removed from that queue. Possibly, if an SOP flag is implemented then if the queue is not empty and the SOP flag is set and the queue head pointer is equal to the GFDP or is between the GFWP and GFDP (empty zone) a packet remove operation for the queue is scheduled.

A system in accordance with the present invention utilizes a global input FIFO which manages the memory as a single FIFO when packets are inserted thereto and manages the memory as a plurality of FIFO queues when removing packets. A memory management scheme in accordance with the present invention takes advantage of the nature of packet switching to give an efficient implementation of multiple independent queues in a memory. The key observation that is made is that in a packet switch it is expected that packets may be stored in the memory for a time no greater than the time it takes to fill up the memory. If a packet is delayed due to extreme congestion, then deleting that packet is an acceptable result.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in

the art without departing from the spirit and scope of the appended claims.